Il Nuovo Sistema di Data Transfer per l'esperimento AMS02

Matteo Jeroen Boschini
CILEA, Segrate

Abstract
Viene descritto il nuovo sistema di trasferimento dati sviluppato dal CILEA e adottato dal gruppo italiano che collabora all'esperimento di astroparticelle AMS02. Il nuovo sistema è basato su un automa agli stati finiti, descritto in XML e implementato in Python, le cui azioni di transizione sono programmi stand-alone. L'uso dell'automa ha permesso la semplificazione del codice e la parallelizzazione delle transizioni. Il sistema è in grado di operare in ambiente *client/server* tradizionale o in ambiente GRID. Il nuovo sistema è in produzione dal marzo del 2007 e ha finora processato 4 TB di dati MonteCarlo.

We present the new Data Transfer System developed by CILEA and adopted by the Italian group that collaborates to the AMS02 astroparticle experiment. The new system is based on finite states automata, described by XML and implemented in Python, whose transition actions are stand alone programs. The use of the finite state automata has greatly simplified the code and has allowed threading. The system is capable to operate in a traditional client/server mode or in a GRID environment.

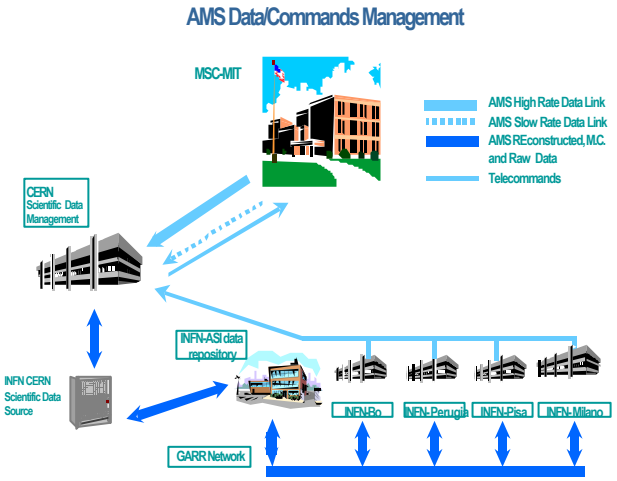
Keywords: AMS02, finite state automata, Python, Data Transfer, Grid

AMS02

AMS02 – *Alpha Magnetic Spectrometer* è un esperimento di astroparticelle volto alla rilevazione di antimateria, materia oscura e materia mancante, che acquisirà dati a partire dal 2009 a bordo della International Space Station (ISS) [1]. Un suo prototipo, AMS01, ha acquisito dati nel 1998 a bordo dello Space Shuttle durante la missione STS-91, ottenendo importanti risultati nel campo della magnetosfera e dei raggi cosmici [2].

Dal 1997 il CILEA partecipa in maniera attiva al progetto AMS [3] [4] in collaborazione con la sezione di Milano-Bicocca dell'INFN (Istituto Nazionale di Fisica Nucleare). In particolare, dal 2001 il CILEA si occupa del sistema di trasferimento dati [5] [6] preposto a trasferire tutti i dati dell'esperimento dal Science Operation Control Center (SOCC), ospitato presso il CERN, all'AMS *Italian Ground segment Data Storage* (IGSDS), presso il CNAF.

Il flusso dei dati dell'esperimento è riportato schematicamente nella seguente figura:



I requisiti del sistema di Data Transfer (DT) sono essenzialmente:

- sostenere un *throughput* di almeno 36 Mb/s (pari a tre volte il data rate previsto);
- implementare un sistema di *book-keeping* su RDBMS;

– essere il più possibile portabile e trasparente alle architetture hardware e software sottostanti.

Il sistema deve inoltre anche poter essere usato per trasferire dati di MonteCarlo, prodotti presso i vari laboratori membri della collaborazione, al SOCC del CERN.

Il Nuovo sistema di Data Transfer

Una prima versione del sistema di Data Transfer, scritta interamente in Perl e in C, è stata usata in produzione dal 2004 a metà del 2007 [7], sia per trasferire i dati dal SOCC all'IGSDS che dai laboratori al SOCC. In totale, tale versione ha trasferito con successo 11 Tbyte, con una consistenza intrinseca[1] pari a 0,002. Tale versione era in grado di funzionare solo in modalità standard client/server ed era difficilmente portabile con trasparenza in ambiente GRID, come richiesto dal CERN e dal CNAF e per ottimizzare l'uso della risorsa di storage HPSS CASTOR [8], usata in entrambi i siti. Inoltre, tale versione non permetteva una semplice parallelizzazione[2] e, sebbene il requisito di 36 Mb/s di throughput fosse ampiamente soddisfatto, si volevano separare tra loro le varie operazioni (validazione input, trasferimento su rete, validazione output, sincronia, book-keeping) per ottenere un throughput più alto.

Questi fattori hanno portato alla decisione di un cambiamento dell'architettura. Tuttavia il meccanismo di book-keeping, basato su RDBMS MySQL [9], *twofold redundant*[3], e quello di *handshake* tra i due siti coinvolti, basato su OpenSSL [10], si è dimostrato più che soddisfacente, sia in termini di prestazioni che di flessibilità, ed è quindi stato mantenuto.

Si è quindi passati anzitutto a Python [11], per poter gestire in maniera robusta la parallelizzazione dei singoli processi.

Il processo di riscrittura in modalità *multi-threaded* ha inoltre reso evidente che tutta l'architettura poteva essere descritta da un automa agli stati finiti. Infatti, semplificando qui per fini di brevità, gli stati nei quali si può trovare un file possono essere:

– da trasferire;
– trasferito;
– trasferito correttamente;
– validato[4].

Le transizioni di stato sono dei semplici programmi che vengono eseguiti come se fossero *stand-alone*, e la condivisione delle informazioni necessarie è gestita da *thread* Python. In effetti le singole transizioni di stato sono effettivamente eseguite da programmi Perl derivati, con minimi cambiamenti, da quelli usati nella versione precedente. È l'automa a stati finiti, scritto in Python, che si incarica di gestire simultaneamente le varie operazioni sui vari file, leggendo da un file XML la descrizione dell'automa e, in particolare, i programmi stand-alone da eseguire per le transizioni di stato. Il sistema, in esecuzione su server gestiti da CILEA presso il SOCC del CERN, essenzialmente esegue i seguenti processi:

– *main,* in ciclo infinito, che legge il catalogo dei file dall'RDBMS Oracle 10i dell'esperimento, presso il SOCC;
– il catalogo viene confrontato con il catalogo MySQL del Data Transfer al SOCC. Se vengono rilevati nuovi file, questi vengono messi in stato TO_BE_TRANSFERED nel database;
– a quel punto, una thread prepara i dati necessari al trasferimento (size, crc, adler32);
– un'altra thread effettua il trasferimento vero e proprio dei file in stato TO_BE_TRANSFERED. Attualmente esso può avvenire, a seconda dei parametri di configurazione, usando *bbftp* [12][5], per la modalità client/server, o il programma *glubus-url-copy,* per la modalità *grid;*
– parallelamente, una thread dedicata, per ogni file TRANSFERED ma non VALIDATED, effettua una connessione (SSL) al DT server DT dell'IGSDS e chiede conferma della ricezione (size, crc, adler32) e, se tutto è avvenuto correttamente, mette il file in stato VALIDATED.
– infine, per ogni file VALIDATED, una thread dedicata effettua una interrogazione a CASTOR all'IGSDS tramite *grid-ftp-ls*[6] per verificare la corretta archiviazione. Se questa non è

[1] Definita come numero di file erroneamente considerati come trasferiti correttamente. Essenzialmente tale "stato" è causato da interruzioni e guasti degli apparati coinvolti.

[2] Essenzialmente ciò è dovuto a una non robusta implementazione delle *POSIX thread* in Perl5 (tutte le versioni fino alla 5.8.X). Infatti, mentre lo *share* di segmenti di memoria allocati a variabili scalari è affidabile, non lo è quello di aree di memoria associate ad *array* o *hash*, o ai loro puntatori.

[3] I database di *book-keeping* sono replicati in modalità *quasi-realtime* presso SOCC, IGSDS e i laboratori INFN di Milano-Bicocca.

[4] Questo stato implica non solo il corretto trasferimento (controllo di *checksum adler32*), ma anche la validazione dal punto di vista del contenuto del file trasferito alla luce di analisi di astrofisica e la corretta scrittura su CASTOR.

[5] Opportunamente modificato nelle parti di autenticazione e uso del parametro TCP_WIN_SIZ.

[6] O una connessione

avvenuta, il file viene rimesso in stato TO_BE_TRANSFERED.

Il vantaggio dell'uso di processi indipendenti è qui evidente: se le singole operazioni venissero effettuate in serie, il sistema avrebbe come throughput massimo quello del processo più lento, mentre con l'uso di thread parallele e indipendenti, il sistema è comunque sempre in grado di ottenere la performance più alta da ogni singolo passaggio.

A dimostrazione di quanto affermato, basti sapere che, mentre il vecchio sistema era in grado di ottenere un throughput generale di 36,4±2,1 Mb/s, dominato essenzialmente dai processi di pre e post-validazione, il nuovo sistema ha un throughout[7] complessivo di 58,6±1,6 Mb/s.

Il sistema attuale è in produzione dal maggio 2007 e ha trasferito correttamente[8] 3500 file, per un totale di 3,1 Tbyte.

È interessante notare che i file non validati correttamente al primo trasferimento, pari a circa il 2% del totale, sono quelli per i quali è fallita la scrittura su CASTOR[9]. Tali file sono comunque correttamente ritrasferiti non appena risolti, tramite procedure automatiche, i problemi su HPSS.

Conclusioni

La nuova versione del sistema di trasferimento dati per AMS02, basata su un automa agli stati finiti, è entrata in produzione nel maggio 2007. Grazie all'uso di thread indipendenti, coordinate tra loro dall'automa agli stati finiti, il throughput medio ottenuto è di 58,6±1,6 Mb/s, altamente superiore ai 36 Mb/s richiesti.

Il meccanismo di book-keeping, solo leggermente modificato rispetto alla versione precedente, ha confermato la sua affidabilità e robustezza. Infine, il nuovo sistema implementa in maniera trasparente l'uso di protocolli GRID, in vista della acquisizione dati di AMS02 a bordo della *International Space Station,* a partire dalla seconda metà del 2009.

Appendice

Di seguito alcuni dettagli tecnici sul sistema.

[7] Calcolato sul trasferimento di 3500 file, per un totale di 3,1 Tbyte.

[8] Ovvero, con validazione finale corretta.

[9] Una più approfondita analisi ha mostrato come tali eventi siano legati ai tentativi di riscrittura su CASTOR_STAGE di file di cui precedentemente, per cause legate a guasti e/o manutenzioni hardware del sistema HPSS, era fallita la scrittura.

- Sistemi operativi supportati: Scientific Linux CERN 3.x e 4.x, Debian 3.x.
- RDMBS supportati: MySQL 3.x, 4.x, Oracle 10i
- Ambiente grid: 3.1 gLite middleware
- HPSS supportati: CASTOR 1 e 2

Bibliografia

[1] M. Aguilar et al., "The Alpha Magnetic Spectrometer (AMS) on the International Space Station, Part I, Results from the test flight on the Space Shuttle" , *Physics Reports*, *366/6* (Aug.2002).

[2] J. Alcaraz, B. Alpat, "Helium in near Earth orbit", *Phys Lett B 494: (3-4)*, Nov. 30, 2000.

[3] M. Boschini, L. Trombetta, "Un DataBase Oracle per l'esperimento AMS", *Bollettino del CILEA*, n. 64, settembre 1998.

[4] Boschini, M. et al., "The AMS-1 Milano Data Center", *Nuclear Science Symposium, 1999. Conference Record.* 1999 IEEE Volume 1, 1999.

[5] M. Boschini, "La gestione dei dati per l'esperimento AMS", *Bollettino del CILEA*, n. 68, giugno 1999.

[6] M. Boschini et al., "AMS-02 Italian Data Transfer", *8th International Conference, ICATPP 2003: Astroparticle, Particle, Space Physics, Detectors and Medical Physics Applications,* World Scientific, 2004.

[7] M. Boschini et al.,"AMS02 Italian Data Transfer System: Real life Experience", *9th International Conference, ICATPP 2005: Astroparticle, Particle, Space Physics, Detectors and Medical Physics Applications*, World Scientific, 2006.

[8] URL: http://castor.cern.ch

[9] MySQL AB. URL: http://www.mysql.com

[10] URL: http://www.openssl.org

[11] URL: http://www.python.org

[12] URL: http://doc.in2p3.fr/bbftp/